# APX

ATARI® PROGRAM EXCHANGE

Bob Fraser

INSOMNIA (A SOUND EDITOR)

APX-20073

User-Written Software for ATARI Home Computers

Bob Fraser

INSOMNIA (A SOUND EDITOR)

APX-20073
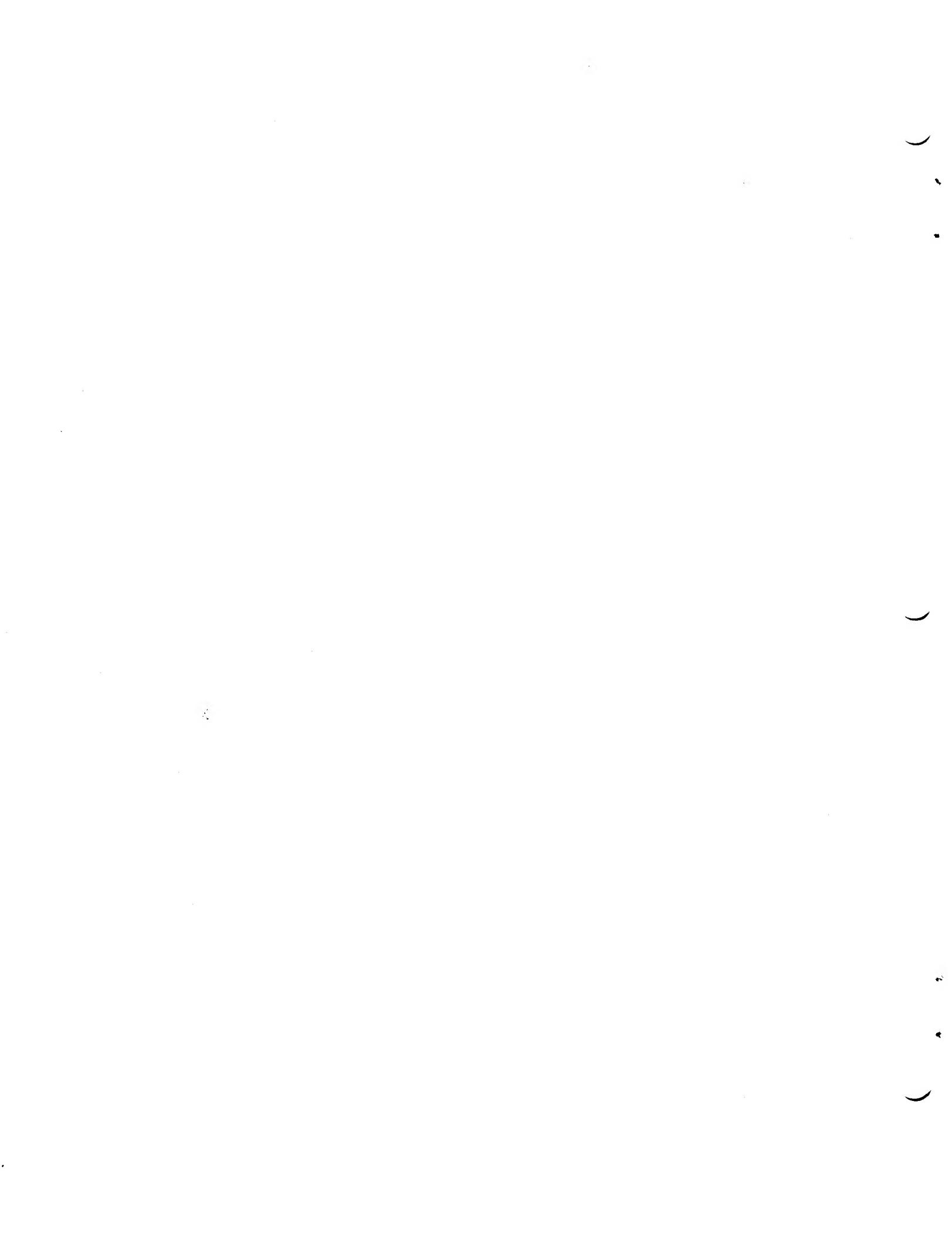
INSOMNIA (A SOUND EDITOR)

by

Bob Fraser

Program and Manual Contents © 1981 Atari, Inc.

TRADEMARKS OF ATARI

The following are trademarks of Atari, Inc.

ATARI®
ATARI 400™ Home Computer
ATARI 800™ Home Computer
ATARI 410™ Program Recorder
ATARI 810™ Disk Drive
ATARI 820™ 40-Column Printer
ATARI 822™ Thermal Printer
ATARI 825™ 80-Column Printer
ATARI 830™ Acoustic Modem
ATARI 850™ Interface Module
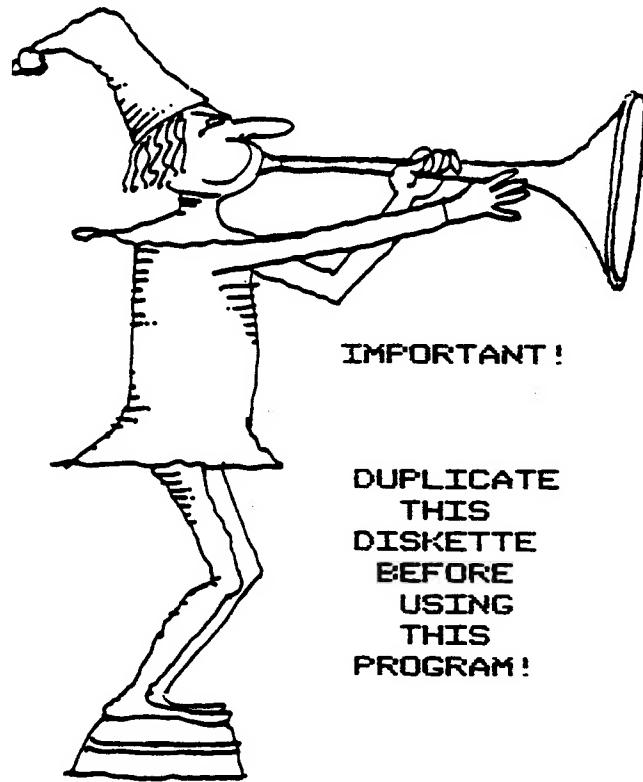
IMPORTANT!

**DUPLICATE
THIS
DISKETTE
BEFORE
USING
THIS
PROGRAM!**

This APX diskette is unnotched to protect the software against
accidental erasure. However, this protection also prevents a program
from storing information on the diskette. The program you've
purchased involves storing information. Therefore, before you can use
the program, you must duplicate the contents of the diskette onto a
notched diskette that doesn't have a write-protect tab covering the
notch.

To duplicate the diskette, call the Disk Operating System (DOS) menu
and select option J, Duplicate Disk. You can use this option with a
single disk drive by manually swapping source (the APX diskette) and
destination (a notched diskette) until the duplication process is
complete. You can also use this option with multiple disk drive
systems by inserting source and destination diskettes in two separate
drives and letting the duplication process proceed automatically.
(Note. This option copies sector by sector. Therefore, when the
duplication is complete, any files previously stored on the
destination diskette will have been destroyed.)

CONTENTS

INTRODUCTION

OVERVIEW

INSOMNIA gives you an easy method for generating a complex (four-voice) sound or a few simple sounds lasting up to four seconds. Then, you can install the sounds in your BASIC programs through a machine language subroutine included in INSOMNIA that lets your program run unimpeded by the sounds. You create sounds by plotting points in three windows of a screen display: frequency, volume, and distortion. You select these sound parameters and plot your points using a joystick controller, a paddle controller, or both. Working with any of the ATARI computer's four sound generators, you immediately hear the changed sound each time you plot a point. You can play your sound either continuously or once only, and you can isolate one voice for modification or you can combine voices to hear the overall effect. INSOMNIA also can produce very low and very high frequency sounds normally unattainable with conventional sound generating methods. You can save and recall sounds on diskette for further revision.

REQUIRED ACCESSORIES

24K RAM
ATARI 810 Disk Drive
ATARI BASIC Language Cartridge
One ATARI Joystick Controller
One ATARI Paddle Controller

GETTING STARTED

LOADING INSOMNIA INTO COMPUTER MEMORY

1. Insert the ATARI BASIC Language Cartridge in the (Left Cartridge) slot of your computer.

2. Insert your ATARI Joystick Controller in the first (leftmost) controller jack at the front of your computer.

3. If you'll be using an ATARI Paddle Controller, insert it in the third controller jack at the front of your computer.

4. Turn on your disk drive.

5. When the BUSY light goes out, open the disk drive door and insert the INSOMNIA diskette with the label in the lower right-hand corner nearest to you. Close the door.

6. Turn on your computer and TV set.

7. When the READY prompt displays on your TV screen, type RUN "D:INSOMNIA.ED" and press the RETURN key. The program will load into computer memory and start.

THE FIRST DISPLAY SCREEN

After a short delay, the program's graphics screen appears and the editor is awaiting your commands.

USING INSOMNIA

INTRODUCTION

INSOMNIA is a powerful sound editing package for creating sound effects for your BASIC programs. You create sounds using the program's editor. Then you append a small subroutine to your BASIC program to play the sounds. INSOMNIA's techniques have distinct advantages over the conventional trial-and-error approach to using FOR-NEXT loops and SOUND and DATA statements. The editor eliminates the tedious practice of modifying programs to change sounds. Moreover, the subroutine that plays the sounds is in machine language, requiring no BASIC calls or intenvention and virtually no execution time. In fact, the subroutine lets the main BASIC program execute unhindered while perhaps a bird chirps in the background! The speed gained is substantial. Normally, a BASIC program must use all its speed resources to create interesting sounds (using FOR-NEXT loops, SOUND statements, and so on.). A BASIC program using INSOMNIA, however, performs interesting sounds with the time expenditure of only an occasional POKE. This means that BASIC is free to process while sounds play, something not possible with the conventional approach.

This manual consists of three parts: (1) using the editor, (2) putting the sounds into a BASIC program, and (3) operation. The first section contains step-by-step instructions on how to use the editor portion of INSOMNIA. Read through this section once as a tutorial and use it as a reference thereafter. Before proceeding with the first section, study the following screen diagram, and refer back to it while reading; it should help clarify the explanations. Don't expect to emerge from this session able to produce sounds quickly. Becoming familiar enough with INSOMNIA just to sit down and create a desired sound takes practice.

The second section contains instructions for putting the created sounds to work in your BASIC programs. It's fairly straightforward. Once you've created sounds, it should be only a matter of minutes before your program is using them.

The third section covers the techniques used to produce the sounds, file structures, and miscellaneous technical information. It's intended for the curious and advanced programmer.

SUB-WINDOW SELECTORS - THE LIT ONE SHOWS WHICH SUB-WINDOW WILL BE AFFECTED BY THE PADDLE AND JOYSTICK. CONTROLLED BY 'SELECT' KEY AND JOYSTICK BUTTON.

VOICE PLAYING INDICATORS - ANY OF THESE LIT MEAN THAT THAT VOICE IS PLAYING. CONTROLLED BY 'P', 'T'.

VOICE SELECTION POINTERS - THESE INDICATE WHICH VOICE WILL RECEIVE THE NEXT COMMAND. CONTROLLED BY '1', '2', '3', & '4'.

DOT INDICATES WHICH VOICE IS CURRENTLY DISPLAYED IN THE WINDOW. CONTROLLED BY 'E' COMMAND.

THESE TWO NUMBERS TELL WHICH 2 SECONDS OF THE SOUND ARE BEING DISPLAYED (SOUNDS CAN BE 4 SECS, BUT THE WINDOW ONLY HOLDS 2). CONTROLLED BY '>', '<'.
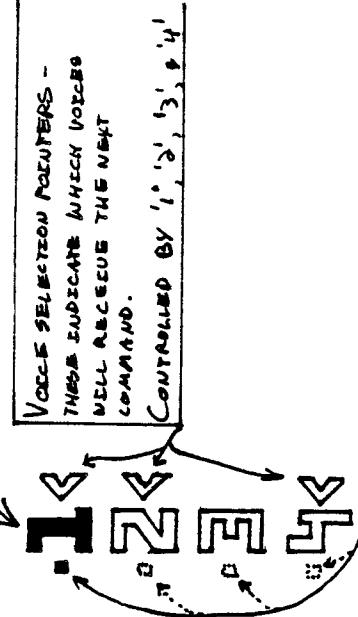
END OF SOUND INDICATOR - MARKS THE END OF THE USED SOUND DATA. CONTROLLED BY 'R', 'O'.

CURSOR - DOT BETWEEN BARS IS THE ONE CURRENTLY RECEIVING ACTION. CONTROLLED BY JOYSTICK AND PADDLE TRIGGER.

FREQUENCY DATA - THE HIGHER THE DOT, THE HIGHER THE NOTE.

DOTS ARE MOVED BY LIGHT PEN, PADDLE, AND JOYSTICK

VOLUME DATA - THE HEAVIER THE DOT, THE LOUDER.

SUB-WINDOWS - THESE HOLD FREQUENCY, VOLUME, DISTORTION SOUND DATA.

COLLECTIVELY REFERRED TO AS THE 'WINDOW'.

DISTORTION DATA - BOTTOM IS DISTORTION Ø, TOP IS DISTORTION 14

ENOUGH ROOM FOR 120 DOTS, OR 2 SECONDS.

Figure 1   Editor Screen Diagram

-4-

USING THE EDITOR

CONTROLLERS AND GRAPHICS SCREEN

The editor uses whichever controllers you've plugged into your computer console (the joystick controller belongs in the first jack, the paddle controller in the third jack). Hold the joystick and paddle controllers so that the red trigger buttons are at your upper left.

The editor's graphics screen is very straightforward. Figure 1 is a schematic of the editor screen.

Three boxes comprise most of the screen. These are referred to as subwindows. The uppermost, largest subwindow contains frequency data, the middle subwindow volume data, and the lowest subwindow distortion data. The terms "frequency", "volume", and "distortion" should sound familiar since they're the last three parameters of the ATARI BASIC "SOUND" statement. The SOUND statement's frequency parameter determines tone, the volume parameter controls the volume of the resulting sound, and the distortion parameter controls the noise content. The editor interprets these three parameters in precisely the same manner as ATARI BASIC interprets them in a SOUND statememt. In fact, the editor's windows define the equivalent of 120 successive SOUND statements.

EDITOR COMMANDS AND CONTROLLER MOVEMENT

Descriptions of each command follow. To use a command, press the underlined letter(s).

A--Add length

> Before any other command will work on an empty window, you must press the "A" key several times, or hold it down for a period of time. A vertical blue bar moves to the right slightly with every press. As it moves to the right, it opens a space to the left in which you can create and edit your sound. The blue bar defines the length of the sound. Each movement of the bar reflects a sixtieth of a second increase in the duration of the sound. The window contains 120 dots horizontally; thus, two seconds of the sound display in the window.

D--Delete length

> This command is the opposite of the "A" command. Every press of the "D" key decreases the duration of the sound in the window by one-sixtieth of a second, and moves the vertical blue bar one space to the left.

> Use the "A" and "D" keys to move the blue bar to the middle of the screen.

JOYSTICK MOVEMENTS

Move the joystick to the right and then to the left. A vertical red double bar moves across the screen in the direction you move the joystick. This double bar is called the cursor. Press the joystick's red trigger button repeatedly. Each press alternately lights an "F", "V" or "D" next to one of the three subwindows. The lighted letter indicates the editor's mode of operation; i.e., when the "F" lights up, all joystick and paddle operations affect the upper window, the window corresponding to the letter "F". When the "V" or "D" lights up, the controllers affect the middle (volume) window or the lower (distortion) window. Press the trigger until the "V" lights up. Then push the joystick up and then down. The dot between the cursor's double bars breaks away from the other dots and moves in the direction in which you push the joystick. Push the joystick right once (moving the cursor one position to the right) and move the next dot up in the same manner. As the dots rise, a periodic hissing sound begins repeating. Since two dots have been raised, each having a duration of one-sixtieth of a second, the hiss lasts for two-sixtieths, or one-thirtieth, of a second. Since the vertical blue bar has approximately sixty spaces to its left, the thirtieth of a second hiss repeats once per second.

PADDLE CONTROLLER

The paddle allows faster updates in many cases. Use the SELECT button on the computer keyboard to select the subwindow desired. The paddle position then determines the vertical position of the dot under the cursor. The paddle trigger moves the cursor to the right one step. This controller is initially more difficult to use, but with practice is handier.

To set all the volumes to a constant, the fastest method is as follows: set the paddle to the desired volume level, and then press the trigger button. The cursor will begin moving to the right bringing all the dots to the first one's level.

If the paddle isn't fully to the left, the dot under the cursor blinks steadily, indicating that the paddle is continuously updating the dot. This can hinder other controller operations, so whenever you're not using the paddle, turn it completely to the left.

LS--Load Sound

Use LS to load a sound file you've saved previously using the "SS" command--see below). Once loaded, you may edit the file as desired.

Press the "L" key and then the "S" key. The graphics screen dissappears and a message displays requesting you to type in a filename from which to load the sound. Type "D:DEMO.SND", and press the RETURN key. When the demonstration sound is done loading, you'll hear a simulation of a steam engine. The display shows the frequency starting high and dropping low, volume set to a constant 8, and a constant distortion setting of 0. To make the engine speed up, press the "D" key several times.

1, 2, 3, and 4 keys--Voice selection

Although we've experimented with only voice 1, the editor works with all four of the computer's sound generators. A number of the editor's functions need to know on which of the four voices to perform the function. Use the first four number keys to tell the editor the voices you want to work on.

Press the "1" key; a small pointer appears the right of the "1" in the upper right corner of the screen. Press the "3" and the "4"; similar pointers appear next to the "3" and "4". These pointers indicate that you've selected voices 1, 3, and 4. To deselect voices, press the appropriate keys again. For example, perhaps you meant to select only voice 1 in the above example; press the "3" and the "4" keys, and the small pointers next to those numbers will disappear.

C--Clear voice selections

Use "C" as a quick way to clear all your voice selections with one keystroke. You can accomplish the same thing by pressing the numbers of the voices you want to clear.

T--Trigger selected voice

With voice 1 selected as above, press the "T" key; the steam engine stops. Until you pressed "T", the "1" in the upper right corner of the TV screen was lighted, indicating that voice 1 was playing continuously. "T" thus stops the selected voice(s) and dims the corresponding indicator(s) to show that the voice is no longer playing.

You can also use "T" to play the voice through once and then stop. This is useful when you don't want the sound to play repeatedly, such as an explosion. Press the "1" key then the "T" key; a single chug from the steam engine will play.

P--Play selected voice

"P" is the opposite of "T". Press the "1" key followed by the "P" key; the steam engine begins chugging continously, as it did before you used the "T" command to stop it. Familiarize yourself with the editor's controls by using the "P" and "T" commands to play various combinations of the four voices.

E--Edit selected voice

As mentioned earlier, the editor works with any of the four voices. From experimenting with the "P", you probably discovered that the other voices in the demonstration sound also contain sound effects. To put them on the screen so that you can view and edit them, use the "E" command.

Press "2" and then "E"; the steam engine display clears, replaced by another display, that of voice 2. In this way, you can access any of the four voices.

A small dot to the left of the voice number in the upper right corner of the screen indicates the voice currently being edited (on the screen). The dot is currently to the left of the "2".

<u>LO</u>--Change to LOw tone)

    While a voice is playing, press "L" and then the letter "O"; the tone will drop
dramatically. This option lets you create very low frequency sounds, sounds that
normally aren't possible. It works by setting the lowest bit in AUDCTL (location
53768), which changes the clock base from 64 Khz to 15 Khz.

<u>HI</u>--Change to HIgh tone

    This command is the opposite of "LO". It changes the clock base from 15 Khz back to
64 Khz; the result is a higher tone. The change is accomplished by clearing AUDCTL's
lowest bit.

<u>R</u>--Resynchronize selected voices

    Use this command only when you want two or more voices to produce a single sound.
If you play these voices continuously and edit them, they'll probably become out of
phase with each other. Pressing "R" causes all your currently selected voices to
start at their beginning.

    This command will be effective only if the sounds involved are of the same length.
If they're not, then the second repetition puts them out of phase again.

<u>SS</u>--Save Sound as a file

    When you've created sounds worth saving, press the "S" key twice. As with the "LS"
command, the graphics screen disappears, and the editor requests the file name under
which you want the sounds in the editor saved. Give a standard file name
specification (You might want to use the extender ".SND" to distinguish the file as
a sound file). The program will write all pertinent sound information to that file
and then resume normal operation.

    As mentioned previously, the windows can contain 120 dots (which works out to two
seconds). The editor, however, allows sounds up to four seconds in duration. The
program automatically allocates four seconds per voice and saves each voice (using
the "SS" command) on diskette as a four-second voice, regardless of its real length.

<u>></u> and <u><</u>--Page forward and backward

    The screen displays two seconds of a sound at a time, but four seconds are available
for each sound. You use these commands to move the window to display the undisplayed
portions (i.e., seconds) of the sound. A "1-2" displays in the lower right corner
of the screen. These numbers indicate the screen currently contains seconds 1 and 2
of the sound. Press the "A" key until the vertical blue bar passes into the right
half of the window, and then press the ">" key. The screen redraws, moving the
right half of the screen to the left, and blanking the right. The lower right corner
of the screen now shows a "2-3", indicating that seconds 2 and 3 are displayed in
the window. Repeat these steps to display seconds 3 and 4.

    As you might guess, the "<" command moves the window back one second, in this case
displaying seconds 1 and 2, as before.

PUTTING THE SOUNDS
INTO A BASIC PROGRAM

Once you've used the editor to create sounds for your BASIC program, the next step is getting these sounds into your program. Part of the INSOMNIA package is a BASIC subroutine to do just that. This subroutine reads in only to the end-of-sound marker, so the memory used in your BASIC program is only as long as your sound, even though the editor program saves each voice on diskette as a four-second sound.

There are three steps to installing a sound. First, append the subroutine INSTALER (in the file named "INSOMNIA.INS") to your program. Be careful not to clobber any of your program. The routine's statement numbers span 30000-30100, so this shouldn't be a problem.

Second, add the following statement to the initialization phase of your program:

        DIM CIO$(39),SET$(10),SNDFILE$(14)

Third, set SNDFILE$ to the name of the the file containing your sounds (the name you typed when you did the "SS" command). Then, to initialize the sound, call the routine at statement 30000. For example:

        SNDFILE$="DEMO.SND":GOSUB 30000

Once the call is executed, the sounds are at your program's disposal. The POKE statement controls them. The following table illustrates the memory locations used, and their function:

TABLE 1   Memory Locations and Their Functions

Location	Name	Description	Comments
1536	TRIG1	trigger voice 1	POKEing a one in these locations
1537	TRIG2	trigger voice 2	causes the voice to play once,
1538	TRIG3	trigger voice 3	providing it's in trigger mode
1539	TRIG4	trigger voice 4	(see below)
1540	MODE1	mode of voice 1	POKEing a 0 here causes the voice
1541	MODE2	mode of voice 2	to play continously.
1542	MODE3	mode of voice 3	A 1 means "trigger" mode. Whenever
1543	MODE4	mode of voice 4	a one is POKEd in TRIGx, it plays once.
1544	COUNT1	length of voice 1	These store the lengths of the
1545	COUNT2	length of voice 2	voices in 60ths of a second. They're
1546	COUNT3	length of voice 3	normally not modified, except for
1547	COUNT4	length of voice 4	example, making an engine go faster.
1548	PTR1	current note v1	These hold which 60th sec note is
1549	PTR2	current note v2	currently being played. It ranges
1550	PTR3	current note v3	from 0 to COUNTx.
1551	PTR4	current note v4	
1552,53	FPTAB1	freq tabl ptr v1	These are double-byte addresses of
1554,55	FPTAB2	freq tabl ptr v2	the frequency data tables. Don't
1556,57	FPTAB3	freq tabl ptr v3	alter them in order for
1558,59	FPTAB4	freq tabl ptr v4	them to function properly.
1560,61	CPTAB1	cntl tabl ptr v1	These are double-byte addresses of
1562,63	CPTAB2	cntl tabl ptr v2	the audio control data tables. Don't
1564,65	CPTAB3	cntl tabl ptr v3	alter in order for them
1566,67	CPTAB4	cntl tabl ptr v4	to function properly.

To understand using these control registers, load INSTALER and add two lines of BASIC:

```
1000  SNDFILE$="DEMO.SND":GOSUB30000
1010  INPUT X,Y:POKE X,Y:GOTO 1010
```

This program accepts two numbers, and POKEs the second into the first. Use this to become familiar with operating INSOMNIA's sound. Try, for example:

```
?1541,0   --> to play voice 2
?1541,1   --> to turn off voice 2
?1537,1   --> to play voice 2 once
?1537,1   --> to again play voice 2 once
?1540,1   --> to turn off voice 1
```

A FEW WARNINGS

1. INSTALER uses variables CIO$, SNDFILE$, SET$, X, E, S, HI, and LO. If your program uses any of these, you could have a problem. Don't use any of these prior to calling INSTALER.

2. INSTALER clears the screen as part of its required procedure. Be sure to make all GRAPHICS commands as well as screen set ups execute after INSTALER has executed.

3. INSTALER uses IOCB #6, so (again) don't do anything with IOCB #6 until INSTALER has been called.

4. INSTALER stops if any I/O errors occur, or if the file specified isn't a sound file.

5. INSTALER moves the top of memory. Any top of memory calculations in your program should be relative; i.e., don't assume that the top of memory is any particular place. Use PEEK(106) to find the high address.

6. Whenever your program uses the serial bus (i.e., disk or printer I/O), the sounds will not recover. So, following every I/O call, execute a null sound statement (SOUND 0,0,0,0) to re-initialize the sounds.

OPERATION

The key to the success and power of the INSOMNIA package is a machine language program transparent to the user. Called VBSOUND, this program is in the file "VBSOUND.6". It's implemented as a Vertical Blank Interrupt (VBI) service routine, which means that it executes precisely every sixtieth of second along with the Operating System's VBI service routine. This implementation allows the sound to be processed automatically, and independent of the program executing.

VBSOUND is installed as the deferred interrupt routine using the machine language routine in SET$. It can also be removed using SET$; see the editor code for more information on how to do this.

For more details in a tutorial approach to interrupt sound processing, see Chapter 10 of De Re Atari (APX-90008), by Chris Crawford, 1981.

The sound files created by the editor are standard binary format files (see the Disk Operating System II Reference Manual for more information). The first section of every sound file stores the program VBSOUND along with all of its control tables (modes, pointers, and so on). The second section stores the actual sound data.

When installed by INSTALER, VBSOUND resides in page 6, and the screen memory area is moved down, making room for the sound data written at the top of memory.

LIMITED WARRANTY ON MEDIA AND HARDWARE ACCESSORIES.

We, Atari, Inc., guarantee to you, the original retail purchaser, that the medium on which the APX program is recorded and any hardware accessories sold by APX are free from defects for thirty days from the date of purchase. Any applicable implied warranties, including warranties of merchantability and fitness for a particular purpose, are also limited to thirty days from the date of purchase. Some states don't allow limitations on a warranty's period, so this limitation might not apply to you. If you discover such a defect within the thirty-day period, call APX for a Return Authorization Number, and then return the product along with proof of purchase date to APX. We will repair or replace the product at our option.

You void this warranty if the APX product: (1) has been misused or shows signs of excessive wear; (2) has been damaged by use with non-ATARI products; or (3) has been serviced or modified by anyone other than an Authorized ATARI Service Center. Incidental and consequential damages are not covered by this warranty or by any implied warranty. Some states don't allow exclusion of incidental or consequential damages, so this exclusion might not apply to you.

DISCLAIMER OF WARRANTY AND LIABILITY ON COMPUTER PROGRAMS.

Most APX programs have been written by people not employed by Atari, Inc. The programs we select for APX offer something of value that we want to make available to ATARI Home Computer owners. To offer these programs to the widest number of people economically, we don't put APX products through rigorous testing. Therefore, APX produts are sold "as is", and we do not guarantee them in any way. In particular, we make no warranty, express or implied, including warranties of merchantability and fitness for a particular purpose. We are not liable for any losses or damages of any kind that result from use of an APX product.
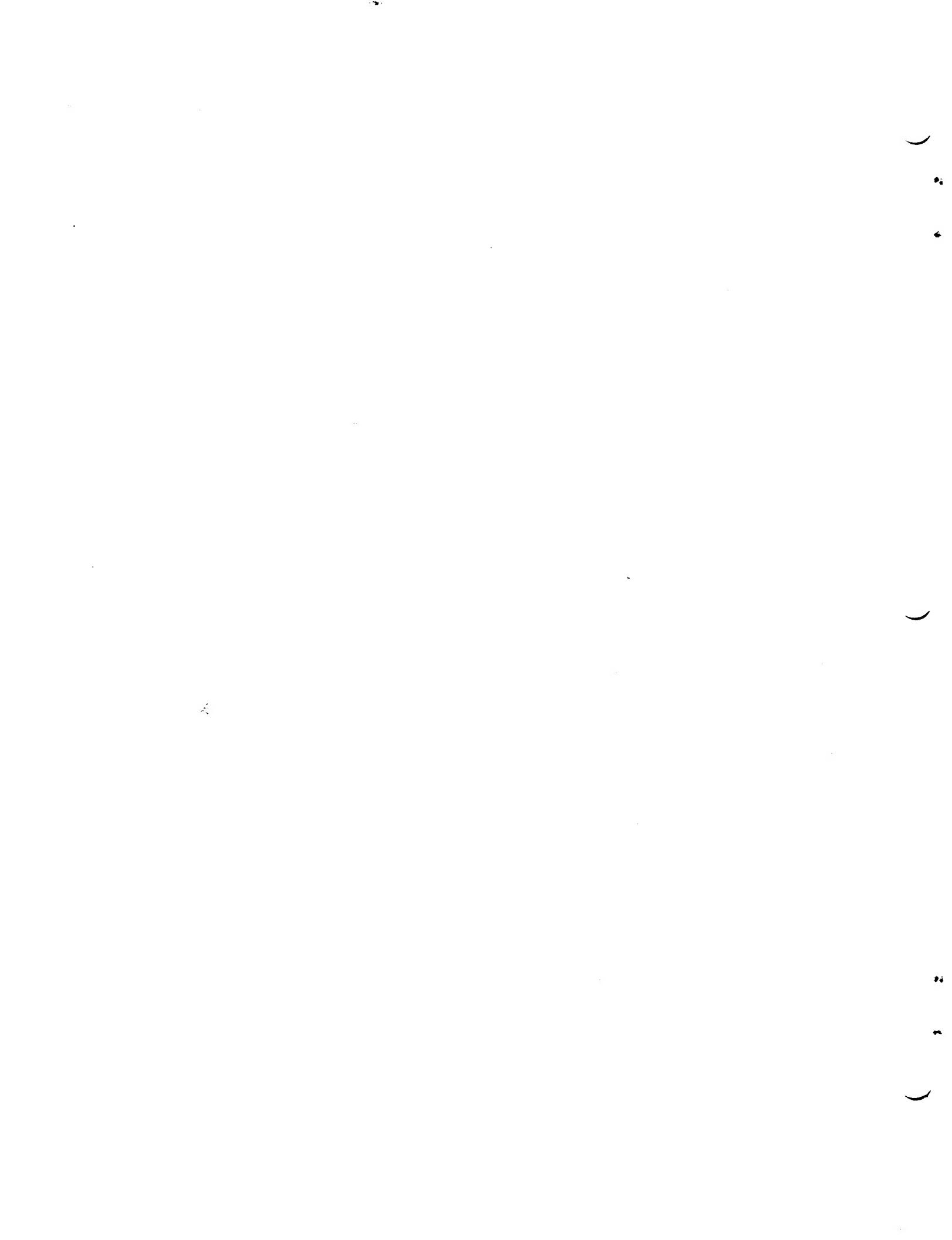
We're interested in your experiences with APX programs and documentation, both favorable and unfavorable. Many software authors are willing and eager to improve their programs if they know what users want. And, of course, we want to know about any bugs that slipped by us, so that the software author can fix them. We also want to know whether our documentation is meeting your needs. You are our best source for suggesting improvements! Please help us by taking a moment to fill in this review sheet. Fold the sheet in thirds and seal it so that the address on the bottom of the back becomes the envelope front. Thank you for helping us!

1. Name and APX number of program _____

2. If you have problems using the program, please describe them here.

_____

_____

_____

3. What do you especially like about this program?

_____

_____

_____

4. What do you think the program's weaknesses are?

_____

_____

_____

5. How can the catalog description be more accurate and/or comprehensive?

_____

_____

6. On a scale of 1 to 10, 1 being "poor" and 10 being "excellent", please rate the following aspects of this program?

_____ Easy to use
_____ User-oriented (e.g., menus, prompts, clear language)
_____ Enjoyable
_____ Self-instructive
_____ Useful (non-game software)
_____ Imaginative graphics and sound

7. Describe any technical errors you found in the user instructions (please give page numbers).

_____

_____

_____

8. What did you especially like about the user instructions?

_____

_____

_____

9. What revisions or additions would improve these instructions?

_____

_____

_____

10. On a scale of 1 to 10, 1 representing "poor" and 10 representing "excellent", how would you rate the user instructions and why?

_____

_____

11. Other comments about the software or user instructions:

_____

_____

_____

_____

_____

_____

```
 -----
|     |
|STAMP|
|     |
 -----
```

ATARI Program Exchange
Attn: Publications Dept.
P.O. Box 50047
60 E. Plumeria Drive
San Jose, CA 95150

[seal here]